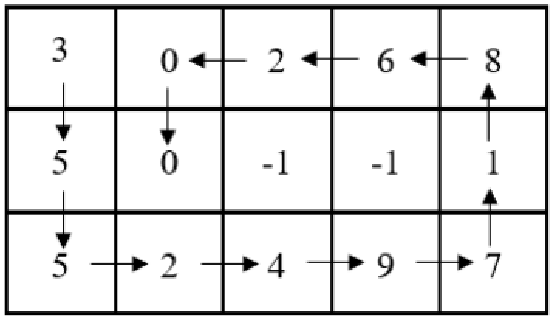
PROGRAMMING TASK

Q1: You must use Linked List for solution of this question. Your code will be tested using .txt files, design your code considering this.



3	0	2	6	8
5	0	-1	-1	1
5	2	4	9	7

A. Return the elements of the given matrix in spiral form. The program should stop when it sees -1. The starting point will be 3 and the direction will be down until the end of the column, then right until the end of the row, up to the top and left.

The sequence should be as follows:

3→5→5→2→4→9→7→1→8→6→2→0→0

Example Input/Output:

```
Input filename:matrix.txt
[3, 5, 5, 2, 4, 9, 7, 1, 8, 6, 2, 0, 0]
```

B. The lists will be given as input, for example:

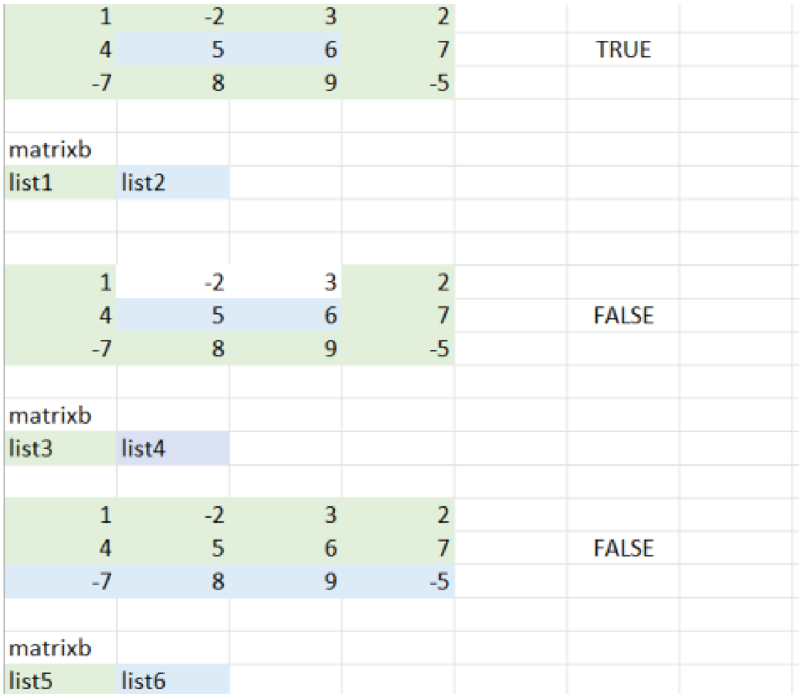
List 1: {3, 5, 5, 2, 4, 9, 7, 1, 8, 2, 6, 2, 0, 0}
List 2: {0, -1, -1}

You need to find out if the neighborhood of list 1 contains list 2 or not. To do this, you need to check whether any node from list 1 is connected to any node in list 2. The nodes to the right, left, under, and above should all be checked. (Checking diagonally is not necessary.)

Example Input/Output:

```
Input filename:matrixb.txt
Input filename:list1.txt
Input filename:list2.txt
TRUE
```

The direction of the arrow is the same as in 1A. You need to read 3 txt files, the first is the matrixb, the second and third are the lists with the indices of the matrices. The output will be TRUE or FALSE.



1	-2	3	2			
4	5	6	7		TRUE	
-7	8	9	-5			
matrixb						
list1	list2					
1	-2	3	2			
4	5	6	7		FALSE	
-7	8	9	-5			
matrixb						
list3	list4					
1	-2	3	2			
4	5	6	7		FALSE	
-7	8	9	-5			
matrixb						
list5	list6					

You must use Linked List for solution of this question!

Q2: You must use stack/queue for the solution of this question! Your code will be tested using .txt files, design your code considering this.

Implement a Java solution that calculates the range of bitcoin values for the current day, given the series of n daily values of this coin.

Range definition:

The maximum number of consecutive days (starting today and going backwards) that the bitcoin value was higher than or equal to today's value.

For example, if the value of a bitcoin over the next 6 days were 90, 65, 70, 60, 75, 80, then the range returned would be 1, 2, 1, 4, 1, 1.

Example Input/Output:

```
Input filename:price2.txt
[90, 65, 70, 60, 75, 80]
[1, 2, 1, 4, 1, 1]
```

Explanation:

90 → return 1 because it is the first element of the range
65 → return 2 because 65 is smaller than 90
70 → return 1 because 90 is greater than 70
60 → return 4 because the last 4 values (60,70,65,90) were higher than or equal to today's value
75 → return 1 because 90 is greater than 75
80 → return 1

You must use stack/queue for the solution of this question!